the Availability Digest

Failure State Diagrams - Hardware/Software Faults Revisited

February 2008

The Three Rs – A Review

In an earlier article entitled Calculating Availability – The Three Rs, published in December, 2006, we defined three different types of "r"s:

r_h = **repair –** The fault that caused the node to fail must be *repaired*. This usually entails a hardware replacement.

r' = **recovery –** Once the node is repaired, it must be *recovered*. This might require that the software environment be reloaded, the applications started, and the databases be opened, among other tasks. Upon the completion of node recovery, system restoration can begin.

R = **restore –** Once one of the failed nodes is fully recovered, the system has the node complement that it needs to be put back into service. In some cases, service can be *restored* to the users at this time. However, this may not always be true. For example, the failed node's database will often have to be resynchronized with the surviving databases. In some cases, a backlog of manually-completed transactions may have to be entered before system operation can be considered to be restored to its normal state.

There is actually a fourth "r," and that is the time to **return** a node or the system to service. We represent this fourth "r" as follows:

- mtr is the mean time to *return a failed node to service*. It is the nodal repair time plus the recovery time. We will denote this as *r*:

$$\text{mtr} = r = r_h + r'$$

- MTR is the mean time to *return a failed system to service*. It is the time required to return a failed node to service *plus* the system restore time. Assuming parallel repair:

$$\text{MTR} = \text{mtr}/2 + R$$

In the article referenced above, we derived the failure probability of a system that must undergo repair, recovery, and restore operations. For an *n* node singly-spared system undergoing parallel repair with other failed nodes, the probability of system failure was shown to be [see Equation (6) in that paper][1]

$$F = \frac{r/2 + R}{r/2}\frac{n(n-1)}{2}(1-a)^2 = \frac{\text{mtr}/2 + R}{\text{mtr}/2} f(1-a)^2 \qquad (1)$$

[1] See also Failure State Diagrams – Recovery Following Repair, *Availability Digest*; December, 2007.

where

F = the probability of system failure.
a = the availability of a node.
f = The number of ways that two nodes could fail and take the system down (if there are *n* nodes, *f* can be as large as *n*(*n*-1)/2).

Hardware/Software Faults – An Intuitive Approach

In that article, we assumed as indicated above that the mtr for a node involved both node repair and node recovery. However, a node only requires repair if there has been a hardware failure. Many system failures are caused by node failures induced by software bugs, operator errors, or environmental faults. These sorts of failures require only a recovery.

This issue was addressed in our article entitled Calculating Availability – Hardware/Software Faults, published in January, 2007. In that article, we defined

h = probability that a node failure was caused by a hardware fault.

Only *h* of the node failures required repair (*r*), but all node failures required recovery (*r'*). Thus,

$$mtr = hr_h + r' = hr + (1-h)r' \qquad (2)$$

and Equation (1) can be rewritten as

$$F = \frac{[hr + (1-h)r']/2 + R}{[hr + (1-h)r']/2} f(1-a)^2 \qquad (3)$$

This seems to be a very simple and intuitive solution to the hardware/software problem; but as we shall see, this solution indicates the occasional pitfalls of intuitive solutions.

Hardware/Software Faults – A Formal Approach

As we have shown in our series on failure state diagrams,[2] these diagrams can be used to formally derive many of the relationships that we have intuitively derived. In most cases, our intuitive derivation held up to formal analysis.

Let us apply these diagrams to the hardware/software problem. The first step is to create a failure state diagram showing the change in system state leading to dual hardware faults, dual software faults, a hardware fault followed by a software fault, and a software fault followed by a hardware fault. As it turns out, this failure state diagram is so complex that it precludes manual solution.

However, we can take a simplified approach. Given that *h* is the probability of a node failure due to a hardware fault,

- the probability of a dual hardware failure is h^2.
- the probability of a node failure due to a software fault is (1-*h*). Therefore, the probability of a dual software failure is $(1-h)^2$.

[2] Calculating Availability – Failure State Diagrams, *Availability Digest*; September, 2007.
Failure State Diagrams – Repair Strategies, *Availability Digest*; October, 2007.
Failure State Diagrams – Recovery Following Repair, *Availability Digest*; December, 2007.

- the probability of a hardware fault followed by a software fault is *h*(1-*h*).
- the probability of a software fault followed by a hardware fault is (1-*h*)*h*.

These probabilities add up to 1, as expected. Note that the probability of a hardware and a software fault in any order is 2*h*(1-*h*).

We know the probability of a dual hardware fault and a dual software fault from our previous analyses:

$$\begin{aligned} F_{hh} &= h^2 \frac{r/2+R}{r/2} f(1-a)^2 \qquad \text{(dual hardware fault)} \\ F_{ss} &= (1-h)^2 \frac{r'/2+R}{r'/2} f(1-a')^2 \qquad \text{(dual software fault)} \end{aligned} \tag{4}$$

where Equation (1) is used and

F_{hh} = the probability that the system will fail due to a dual hardware fault.
F_{ss} = the probability that the system will fail due to a dual software fault.
a = the availability of a node due to hardware faults.
a' = the availability of a node due to software faults.

For later reference, note that availability, A, mean time between failure, MTBF, and mean time to return to service, MTR are related by

$$(1-A) = 1 - \frac{MTBF}{MTBF+MTR} = \frac{MTR}{MTBF+MTR} \approx \frac{MTR}{MTBF} \quad \text{if MTBF >> MTR}$$

Thus, we can write

$$\begin{aligned} (1-a) &= \frac{r}{mtbf} \\ (1-a') &= \frac{r'}{mtbf} \end{aligned} \tag{5}$$

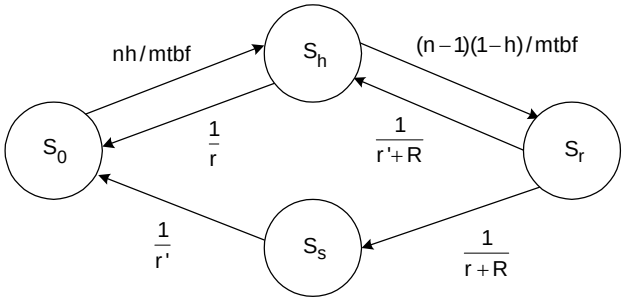
where

mtbf = the nodal mean time between failures.

Now comes the fun part. What about hardware/software failures? To determine the probability of a hardware fault followed by a software fault and the probability of a software fault followed by a hardware fault, we resort to failure state diagrams. We start with the state diagram for a hardware fault followed by a software fault.

In this diagram, *n* is the number of nodes in the system; and the other terms are as defined previously. The states in this diagram are

S_0 – all nodes are operational.
S_h – one node is down due to a hardware fault.
S_s – one node is down due to a software fault
S_r – two nodes are down, the first due to a hardware fault and the second due to a software fault. The system is down.



Hardware Fault Followed by Software Fault

Starting with State S_0, a node could fail either due to a hardware fault or due to a software fault. In this example, we consider only initial hardware faults. Hardware faults occur at a rate of nh/mtbf. That is, each node will fail at a rate of 1/mtbf. There are n nodes that can fail, and *h* of these failures will be due to a hardware fault. Should a hardware fault occur, State S_h is entered; and the system is still operational.

A transition from State S_h can occur for two reasons:

- The node is returned to service. This will occur at a rate of 1/r. The system returns to State S_0.

- A software fault occurs. This will occur at a rate of (n-1)(1-h)/mtbf. State S_r is entered for system repair and recovery, and the system is down.

While in State S_r, one of the two nodes will be repaired; and the system will be recovered and returned to service. There are two ways that this repair can be made:

- The node with the software problem can be repaired and the system recovered, in which case the system returns to State S_h (single hardware failure). This will require a mean time of *r'*+*R* and will occur at a rate of 1/(*r'*+*R*).

- The node with the hardware problem can be repaired and the system restored, in which case the system enters State S_s (single software failure). This will require a mean time of *r*+*R* and will occur at a rate of 1/(*r*+*R*).

When in State S_s, one node has a software failure. State S_0 is entered after a recovery time r'.

The solution of this failure state diagram yields the following for the probability of system failure due to a hardware fault followed by a software fault, F_{hs}:[3]

$$F_{hs} = h(1-h)\frac{n(n-1)}{2}\frac{\left(\frac{r'+R}{r'}\right)\left(\frac{r+R}{r}\right)}{\left(\frac{\bar{r}+R}{\bar{r}}\right)}(1-a)(1-a') \qquad (6)$$

where

$$\bar{r} = \frac{r+r'}{2} \qquad (7)$$

[3] If you want help solving this failure state diagram, contact editor@availabilitydigest.com.

Following this procedure, the probability of a software fault followed by a hardware fault, F_{sh}, yields the same result. Thus,

$$F_{hs} + F_{sh} = 2h(1-h)\frac{n(n-1)}{2}\frac{\left(\frac{r'+R}{r'}\right)\left(\frac{r+R}{r}\right)}{\left(\frac{\bar{r}+R}{\bar{r}}\right)}(1-a)(1-a') \qquad (8)$$

This can be interpreted as the probability of a hardware failure and a software failure in any order that will occur with a probability of 2h(1-h), each failure requiring a single repair, as described above. However, the equation for dual faults has only one term of the form (r+R)/2; and this relationship has two. Therefore, one term has to be divided out; and this is done by dividing by a similar term representing an average repair time.

This equation is intuitively satisfying. It also meets some limiting considerations. If R = 0, then

$$F_{hs} + F_{sh} = 2h(1-h)\frac{n(n-1)}{2}(1-a)(1-a')$$

as one would expect. Also, if r = r', this reduces to

$$F_{hs} + F_{sh} = 2h((1-h)\frac{n(n-1)}{2}\frac{r+R}{r}(1-a)(1-a')$$

as one would expect.

Thus, the probability of system failure due to any pair of hardware or software faults in any order is

$$F = F_{hh} + F_{hs} + F_{sh} + F_{ss}$$

$$= h^2\frac{r/2+R}{r/2}\frac{n(n-1)}{2}(1-a)^2 \qquad \text{(hardware)} \qquad (9)$$

$$+2h(1-h)\frac{n(n-1)}{2}\frac{\left(\frac{r'+R}{r'}\right)\left(\frac{r+R}{r}\right)}{\frac{\bar{r}+R}{\bar{r}}}(1-a)(1-a') \quad \text{(hardware/software)}$$

$$+(1-h)^2\frac{r'/2+R}{r'/2}\frac{n(n-1)}{2}(1-a')^2 \qquad \text{(software)}$$

The mean time to restore the system, MTR, is (assuming parallel repair)

$$MTR = [hr + (1-h)r' + R]/2 \qquad (10)$$

The mean time between failures for the system is

$$MTBF = MTR/F \qquad (11)$$

Comparison of the Intuitive Approach to the Formal Approach

The formal result given by Equation (9) is markedly different from the intuitive approach given by Equation (3). Is there any relationship between these two approaches?

It turns out that these two equations are identical for a restore time, *R*, of zero. This can be seen by using Equations (2) and (5) in Equations (3) and (9) and setting *R* to zero. We also substitute the more general value *f* for the number of failure modes, *n*(*n*-1)/2.

Using Equations (2) and (5), Equation (3) becomes

$$F = f\left[\frac{hr + (1-h)r'}{mtbf}\right]^2 = \frac{f}{mtbf^2}[h^2r^2 + 2h(1-h)rr' + (1-h)^2r'^2] \qquad (3')$$

Using Equations (5), Equation (9) becomes

$$F = f\left[h^2\frac{r^2}{mtbf^2} + 2h(1-h)\frac{rr'}{mtbf^2} + (1-h)^2\frac{r'^2}{mtbf^2}\right] = \frac{f}{mtbf^2}[h^2r^2 + 2h(1-h)rr' + (1-h)^2r'^2] \qquad (9')$$

Therefore, to the extent that the system restore time, *R,* is small compared to the node return-to-service times *r* and *r'*, the intuitive approach holds; and the simpler Equation (3) can be used. If system restore time is large, Equation (9) should be used to calculate system availability.

But just how inaccurate is Equation (3)? Let's look at some examples. We take some reasonable values as follows:

r$_h$ = hardware repair time = 24 hours
r' = node recovery time = node return-to-service time for a software fault = 2 hours
r = node return-to-service time for a hardware fault = *r*$_h$+*r'* = 26 hours
h = probability of a hardware failure = 0.1
n = number of nodes = 8
mtbf = nodal mean time to failure = 4,000 hours

The resultant intuitive errors for various values of *R* are given in the following table:

R (hours)	Intuitive F x 10^5	Formal F x 10^5	Error (%)
0	3.39	3.39	0
0.25	3.77	3.74	0.8
0.5	4.16	4.09	1.7
1	4.93	4.79	2.9
2	6.47	6.15	5.1
4	9.46	8.79	7.5
8	15.7	13.8	13.6

r' = 2 hours, *r*$_h$ = 24 hours, *r* = 26 hours
Accuracy of Intuitive Hardware/Software Failure Relationship [Equation (3)]

Note that the error percent is independent of the number of nodes in the system and the nodal mtbf [see Equations (3') and (9')].

Summary

Over a wide range of reasonable values, the intuitive approach is not only conservative, but it is within 10% of the correct value. Unless *R* is very much larger than either *r* or *r*', the intuitive approach represented by Equation (3) is a reasonable approach to take.

However, if *R* is large compared to either *r* or *r'*, the more accurate expression of Equation (9) for the probability of failure in the presence of hardware and software faults should be used.